手描き曲線からの乱数抽出とそのセマンティクス : 中間報告 (証明論と計算論)

AUTHOR(S):

川西, 暁夫; 鈴木, 登志雄

CITATION:

川西, 暁夫 ...[et al]. 手描き曲線からの乱数抽出とそのセマンティクス : 中間報告 (証明論と計算論). 数理解析研究所講究録 2005, 1442: 8-41

ISSUE DATE:

2005-07

URL:

http://hdl.handle.net/2433/47572

数理解析研究所講究録 1442 巻 2005 年 8-41

8

手描き曲線からの乱数抽出とそのセマンティクス（中間報告）

川西暁夫 (Akio Kawanishi)[1)] ・鈴木 登志雄 (Toshio Suzuki)[2)] *
1): 大阪府立大学 理学系大学院 数理・情報科学専攻／
2): 理学部 情報数理科学科
(Departmet of Mathematics and Information Sciences,
Osaka Prefecture University)
1): whitecat@titan.ocn.ne.jp, 2): toshios@acm.org

平成17年4月5日

概要

川西は鈴木の指導の下で手描き曲線からの乱数抽出を試み，連の検定において優良な成績を示すような抽出法を見出した．このとき抽出された乱数が手描き曲線の不規則性を正しく反映しているかを調べるため，鈴木は Dowd 型ジェネリック・オラクルを用いて「不規則な対象」と「不規則性を保つ写像」の数学的モデルを定義し，その性質を調べた．Dowd 型ジェネリック・オラクルは強制条件の最小サイズを用いて定義される．本稿では「任意の正の整数 r に対して Dowd の意味で r ジェネリックであるオラクル」を「不規則な対象」の数学的モデルとする．そして，川西のアルゴリズムを修正したアルゴリズムを作り，以下の結果を示す．(1) 上記の修正版アルゴリズムは「不規則性を保つ写像」である．(2) 任意の自然数 m に対して，不規則なオラクル D であって，$|\{i \leq n : D(j) = 1\}|/n \to 1/2^m$ (if $n \to \infty$) となるものが存在する．(3) 任意の自然数 m に対して，不規則なオラクル D であって，$|\{i \leq n : D(j) = 1\}|/n \to 1 - 1/2^m$ (if $n \to \infty$) となるものが存在する．(4) 任意の Martin-Löf random oracle は不規則である．(5) さらに，アルゴリズムの出力についての実験結果を示す．

キーワード random number generation, extractor, definition of random sequence, algorithmic information theory, Dowd-type generic oracle, forcing complexity, bitmap, pen tablet.

*The author was partially supported by Grant-in-Aid for Scientific Research (No. 14740082), Japan Society for the Promotion of Science.

1 序

乱数には，確率的アルゴリズムをはじめとして多くの重要な応用がある．また，有限ビット列や（可算）無限ビット列のランダム性の程度をうまく定式化することは数学的に興味深い．本稿の著者の一人 川西は，もう一人の著者 鈴木の指導の下で手描き曲線からの乱数抽出を試み，連の検定（ランダム性についての統計的検定の一種．詳細は後述）などにおいて優良な成績を示すような抽出法を見出した．現在，川西はその抽出方法の実装を進めるとともに，さらに精密な統計的検定に取り組んでいる．

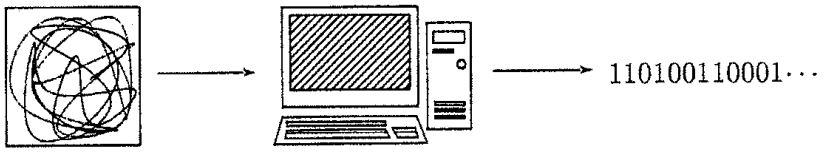

図 1: 手描き曲線からの乱数抽出

ここで問題となるのが，川西が抽出した乱数の統計的性質が，はたして手描き曲線の不規則性を反映した結果なのか，それとも，抽出過程でアルゴリズムが行う計算の結果を主に反映したものなのかということである．つまり，抽出された乱数は乱数源（weak random source）の不規則性を正しく反映しているものなのかを問いたいのである．

単に統計的にある程度よい性質を示す擬似乱数を作るだけであれば，手描き曲線のような乱数源を使わずとも，線形合同法やフィードバック・シフトレジスタなどの方法によって実現できる [Ge03, Kn98]．ただし，これらの方法は一種の漸化式を用いている．そのため，これらの方法によって作られた擬似乱数は「きわめて規則的」である．この「きわめて規則的」という性質をよりフォーマルに言い直すと，コルモゴロフ計算量や回路計算量が低いこととして定式化される．

コルモゴロフ計算量や回路計算量，あるいはその変種のような，不規則性についてのなんらかの数学的指標が川西のアルゴリズムによって保存されているか？もしそうだとすれば，不規則性についてのいかなる数学的指標が保存されているのか？こうした問いに答えるのが本稿の目標である．手書き曲線の外見上の複雑さによって川西のアルゴリズムの出力に違いがあることを示すことは，それなりに興味深いが，それが目標ではない．そうした違いを示しても，与えられた曲線に応じて２種類の線形合同法（より悪い線形合同法と，よりよい線形合同法）を使い分けるタイプのアルゴリズムと川西のアルゴリズムの本質的な違いを示したことにはならない．

上記の課題に答えるため，鈴木は Dowd 型ジェネリック・オラクルを用いて「不規則な対象」と「不規則性を保つ写像」の数学的モデルを定義し，その性質を調べた．そして，川西のアルゴリズムを修正したアルゴリズムを作り，その修正版アルゴリズムが「不規則性を保つ写像」であることを示した．このように乱数の抽出に数学的モデルを与えることによって，抽出された乱数が乱数源の不規則性を正しく反映していることを検証する方法を，プログラムの意味論にならって「乱数抽出の意味論」と呼んでもよいであろう（実装されたプログラムが仕様を正しく反映しているかを調べるため，ハードウェアに依存しない数学的モデルをプログラムに対して与える研究は「プログラムの意味論」とよばれている [vL90]）．

上記の修正版アルゴリズムを用いて抽出した乱数もやはり連の検定について優良な成績を示した．

	プログラムの意味論（例）	乱数抽出の意味論（本稿の場合）
考察の対象	• プログラムの仕様 • 仕様に基づく実装 • 実装されたプログラム	• 乱数源（手描き曲線） • 乱数源からの抽出 • 抽出された乱数
何を知りたいか	実装されたプログラムが仕様を正しく反映しているか	抽出された乱数が乱数源の不規則性を正しく反映しているか
いかなる数学概念で 意味を与えるか	• 位相空間と連続関数（表示的意味論の場合） • 形式的体系と推論（公理的意味論の場合） • 抽象機械と動作（操作的意味論の場合） など	Dowd 型ジェネリック・オラクルおよび， その性質を保つ写像

表 1: プログラムの意味論と乱数抽出の意味論の対比

本稿では鈴木と川西による上記の研究の現状について報告する．本節の残りの部分では，我々のアルゴリズムと意味論についてもう少し詳しく述べ，また，「ほとんどランダムな対象」について他の研究者がこれまで行った定義と我々の定義の比較を行う．

続く第 2 節では用語と記法を説明し，第 3 節で川西の実験について詳細を述べ，

第4節で鈴木による修正版アルゴリズムの詳細を述べる．第5節では，絵の種類（無意味な曲線と漫画風の絵，マウスで描いた絵とペンタブレットで描いた絵）ごとに，川西のアルゴリズムと修正版アルゴリズムの出力を比較した実験結果を示す．第6節で我々の意味論を紹介する．

1.1 川西のアルゴリズムの概要

川西のアルゴリズムにおいて，手描き曲線は正方形のキャンバスをもつモノクロビットマップとして与えられる．ペイントソフトとマウスを用いて，あるいはペンタブレットを用いてこのようなビットマップファイルを作ることができる．適当な直交座標を導入し，各々の座標 (i,j) に対し，その点（ピクセル）がキャンバスと同じ色であるとき (i,j) 成分を0，異なるとき1と約束することにより，ビットマップを正方行列で表すことができる（以下，話を簡単にするため，0を白，1を黒ということもある）．それを仮に行列 $A=(a_{i,j})$ $(i,j\in\{0,\cdots,N-1\})$ としよう．本稿では N を256に固定する．川西のアルゴリズムは，次々にビットマップファイルを受け取りながら，概ね以下の作業を行う．

STEP1: 行列 A の成分を下図のように並べてビット列を作る．

つまり，以下のビット列 a を得る．

$$a=a_{0,0}\cdots a_{0,N-1}a_{1,0}\cdots a_{1,N-1}\cdots a_{N-1,0}\cdots a_{N-1,N-1}$$

STEP2: STEP1で得たビット列 a において，各々の連の長さを2進数で表し，その2進数すべてを連接させて新たなビット列を作る．

注意 1 一般に，$0^i1^j0^k1^\ell\cdots$ (ただし $i\geq 0$, かつ $j,k,\ell,\cdots\geq 1$) という形をしたビット列において，$1^j,0^k,1^\ell\cdots$ の各々を**連**という．$i\geq 1$ の場合は，0^i も連という．このとき，各々の連の長さは i,j,k,ℓ である．

STEP3: STEP2で得たビット列を $b_0\cdots b_{2n-1+k}$ とする．ただし，n は自然数で，k は0または1である．ここで，

$$b_0\,\mathrm{xor}\,b_{2n-1+k},b_1\,\mathrm{xor}\,b_{2n-2+k},\cdots,b_j\,\mathrm{xor}\,b_{2n-j-1+k},$$
$$\cdots,b_{n-1}\,\mathrm{xor}\,b_{n+k}$$

を次々に出力していく．ただし xor は排他的論理和を表す．

そして，次のビットマップファイルを受け取る．以下，繰り返し．

上記の説明においては，詳細を省略している．川西のアルゴリズムのより正確な記述，および検定の結果については，第 3 節で述べる．

1.2 手描き曲線の性質，および数学的モデルへの要請

我々の意味論においては，「不規則な有限ビット列」の数学的モデルを与えず，その代わりに「不規則な（可算）無限ビット列」の数学的モデルを与える．実際のプログラムを通して扱うのは十分長い有限ビット列であるが，理論上はそれを無限ビット列で近似する．有限ビット列全体の集合を $\{0,1\}^*$ で表し，その部分集合を**オラクル**と呼ぶ．オラクルとその特性関数をしばしば同一視する．有限ビット列を短さ優先の辞書式順序を用いて自然数と同一視することにより，オラクルの特性関数は無限ビット列とみなされる．

どのような数学的モデルをつくるべきかを決めるため，ここで手描き曲線の特徴を観察する．

第 1.1 小節で川西のアルゴリズムの概要を示した．手描き曲線は行列 $A = (a_{i,j})$ に変換され，さらにアルゴリズムの STEP1 において行列 A はビット列 a に変換された．我々は多くのビットマップファイル $X_1, X_2, X_3, \cdots$ を作り，それらを次々に行列 $Y^{(1)}, Y^{(2)}, Y^{(3)}, \ldots$ やビット列 $y^{(1)}, y^{(2)}, y^{(3)}, \cdots$ に変換して観察した．その結果，暫定的に以下のような仮説を立てた．

手描き曲線から得られるビット列の性質（経験に基づく仮説）

- 仮説 1: 曲線を描く人が特別な努力をしない限り，各々のビット列 $y^{(j)}$ において，1（黒）の占める比率は 50 パーセントからかけ離れていることが多い．
- 仮説 2: 多項式時間プログラム，多項式サイズ回路族，あるいは簡単な漸化式（たとえば，「$x_{n+k} = f(x_n x_{n+1} \cdots x_{n+k-1})$」，ただし k は自然数で，f は k 変数ブール関数）をあらかじめ用意しておき，それによってビット列 $y^{(1)}, y^{(2)}, y^{(3)}, \cdots$ を予測するのは不可能である．
- 仮説 3: しかし，ビット列 $y^{(1)}, y^{(2)}, y^{(3)}, \cdots$（を連接してできるビット列）の部分列の中には，高い確率で予測可能なものがあり得る．たとえば，曲線を描く人が特別な努力をしない限り，正方形のビットマップの四隅のピクセルは 0（白）になる傾向がある．したがって，ビット列 $y^{(1)}, y^{(2)}, y^{(3)}, \cdots$ の各々の第 0 成分で構成される列 $y_0^{(1)}, y_0^{(2)}, y_0^{(3)}, \cdots$ を作ると，その多くの項が 0 になる傾向がある．

上記の仮説3は，より控えめな仮説で置き換えることができる．たとえば，前処理として正方形のビットマップの周辺部分を切り捨てれば，四隅のピクセルが0（白）になる傾向を抑えることができる．しかし，ビット列 $y^{(1)}, y^{(2)}, y^{(3)}, \cdots$ を連接してできるビット列において，「高い確率で予測可能なビットからなる部分列」がないと考えるのは行き過ぎであろう．

そこで，「不規則なオラクル（無限ビット列）」の数学的モデルを定式化するとき，少なくとも以下の要請1〜3がみたされるようにしたい．

「不規則な対象全体のクラス」の数学的モデルへの要請

以下は，個々の不規則な対象に対する要請ではなく，不規則な対象全体のクラスに対する要請であることに注意されたい．短さ優先の辞書式順序によってすべてのビット列を並べ，この順序において $(i+1)$ 番目のビット列を $z(i)$ で表す．

- 要請1： 任意の自然数 m に対し，不規則なオラクル X であって，極限値

$$\lim_{n\to\infty} \frac{|\{i < n : X(z(i)) = 1\}|}{n} \tag{1.1}$$

が $1/2^m$ 以下のものが存在する．また，不規則なオラクル X であって，極限値(1.1)が $1-1/2^m$ 以上のものが存在する．記号「| |」は cardinality を表す．

（言い換えれば，X が不規則なオラクルであるからといって，極限値(1.1)が $1/2$ に近い値であるとは限らない．）

- 要請2：
  1. 不規則なオラクル X は多項式サイズ回路をもたない（$X \notin \mathrm{P/poly}$）．特に，多項式時間計算可能ではない（$X \notin \mathrm{P}$）．
  2. k は自然数，f は $\{0,1\}^k$ から $\{0,1\}$ への全射，かつ X は不規則なオラクルであるとする．このとき，ある自然数 i が存在して

$$f(X(z(i)), X(z(i+1)), \cdots, X(z(i+k-1))) = 0 \tag{1.2}$$

が成り立つ．

- 要請3：不規則なオラクル X および，疎（sparse）な無限集合 $T \subseteq \{0,1\}^*$ および，T から $\{0,1\}$ への関数 f が存在して以下が成り立つ「T および f は多項式時間計算可能で，T の任意の要素 t に対して $X(t) = f(t)$ が成り立つ」ここで，集合 $T \subseteq \{0,1\}^*$ が**疎**であるとは，ある多項式 p が存在して，任意の自然数 n に対して $|\{u \in \{0,1\}^{\leq n} : T(u) = 1\}| \leq p(n)$ が成り立つことをいう．

（インフォーマルに言えば，不規則なオラクルの定義域を多項式時間計算可能かつ疎な無限集合に制限したとき，規則性をもった関数になる可能性がある．）

1.3 乱数抽出の意味論と主要な結果

この小節では，我々の意味論の根幹の部分を紹介する．

不規則な対象のもつべき性質　オラクル D が

性質 1「任意の自然数 r に対して，D は r-Dowd オラクル（Dowd の意味での r ジェネリック・オラクル）である」

という性質をもつとき，我々は「D が不規則である」と考える．r-Dowd オラクル [Do92, Su01, Su02, Su05] はクエリー（質問）記号付きの命題論理式の体系（*the relativized propositional calculus*）を用いて定義される．The relativized propositional calculus の定義は第 2 節で述べる．ここでは r-Dowd オラクルの定義をインフォーマルに説明する．オラクル X が与えられると，the relativized propositional calculus の式に対して「X に関するトートロジー」という概念が定まる．r を自然数とする．X に関するトートロジーのうち，特に，クエリー記号の出現回数が r 回であるものを「X に関する r クエリー・トートロジー」という．オラクル（の特性関数）の定義域を有限集合に制限して得られる関数を**強制条件**（*forcing condition*）という．

定義 1 [Do92]

1. F がクエリー記号付き命題論理式で，かつ，強制条件 S の拡張になっている任意のオラクル X に対して F がトートロジーになるとき，「S は F を**強制する**」という．
2. オラクル X が *r-Dowd*（*Dowd* **の意味で** r **ジェネリック**）であるとは，ある多項式 p が存在し，X に関する任意の r-クエリー・トートロジー F に対して，ある強制条件 S が存在し，S は X の部分関数であり，S は F を強制し，かつ，S の定義域のサイズ（cardinality）が高々 $p(|F|)$ となることをいう．

任意の正の整数 r に対し，r-Dowd オラクル全体の集合はカントル空間（すなわち，すべてのオラクルからなるクラス）において測度 1 のクラスをなす（証明のあらましは [Do92]，厳密で詳細な証明は [Su01, Su02] を参照されたい）．

不規則性を保つ写像のもつべき性質　カントル空間からカントル空間への計算可能な写像 f が

性質 2「性質 1 をもつ任意のオラクル D に対して，$f(D)$ は性質 1 をもつ」

という性質をもつとき，我々は「f が不規則性を保つ」と考える．カントル空間からカントル空間への写像 f が

性質 2′「任意の自然数 r に対して自然数 s が存在して，任意の s-Dowd オラクル D に対して，$f(D)$ は r-Dowd オラクルである」

という性質をもてば，容易にわかるように，f は性質 2 をもつ．

以下の定理は，後述の定理 8 と並んで，本稿の主要な結果である．

定理（命題 2, 3, 4, 系 6）「オラクルが不規則であるとは，そのオラクルが性質 1 をもつことである」と定めると，小節 1.2 の要請 1,2,3 はすべてみたされる．

上記の定理および性質 1，2 の厳密な定式化は第 6 節において行う．定理 8 の概要は小節 1.5 を，詳細は第 4 節を参照されたい．

手描き曲線から得られた多くのビットマップファイル $X_1, X_2, \cdots$ をビット列 $y^{(1)}$, $y^{(2)}, \ldots$ に変換し，これらを連接してビット列 a を得るとき，a は性質 1 をもった対象であると，本稿では考える．所与のアルゴリズムが a を別のビット列 b に変換するとする．このとき，a の性質 1 が保たれ，なおかつランダム性についての統計的検定に b が合格するならば，その変換アルゴリズムはよいものだと考える．

1.4 「ほとんどランダムな対象」の様々な定義との比較

「ランダムな対象」および「ほとんどランダムな対象」の数学的定義は，これまで多くの研究者によって様々なものが提案されている．この小節では，そうした定義の中の代表的なものをいくつか簡単に復習し，それらに比べて我々の「性質 1」がどのような利点をもっているかを述べる．

Bernoulli sequence and random oracle: 独立な試行を何回か続けて得られる数列はしばしばベルヌーイ列 [Fe68, Fe71] と呼ばれる．0 と 1 が等確率で現れる試行を無限回繰り返して得られるベルヌーイ列を，計算量理論においては，**ランダム・オラクル** [BG81] と呼ぶ．ある性質 φ を持つオラクル全体の集合がカントル空間（の標準的な測度）において測度 1 であるとき，「X がランダム・オラクルであるとき（確率 1 を以って）X は性質 φ をもつ」という．X がランダム・オラクルであるとき（確率 1 を以って）X は Martin-Löf random である（下記を参照）．

Martin-Löf's 1-randomness and Kolmogorov complexity: Martin-Löf random という概念は，ベルヌーイ列の effective version のひとつである．オラクル X が Martin-Löf random であるとは，インフォーマルに言えば，X が effective な統計的検定をすべて通過するということである．以下の定義 2 において，open set とはカントル空間の標準的な位相に関する open set を意味する．

定義 2 [Ml66, YDD04]

1. A computable collection $\{V_n : n \in \mathbb{N}\}$ of computably enumerable open sets is said to be a *Martin-Löf test* if for all n the measure of V_n is at most 2^{-n}.
2. An oracle X is said to *pass* the Martin-Löf test if there exists n such that $X \notin V_n$.
3. An oracle X is said to be *Martin-Löf random* if it passes all Martin-Löf test.

上で定義された Martin-Löf randomness を，*Martin-Löf 1-randomness* ともいう

(Martin-Löf 自身は「ベルヌーイ列」とよんでいる). Martin-Löf randomness について, より正式な取り扱いは [Ca02] を参照されたい. Martin-Löf random の概念は, Kolmogorov complexity [BDG90, LV97] と密接な関係がある. ビット列 u に対し, その (prefix free) Kolmogorov complexity を $K(u)$ で表す. オラクル X が Martin-Löf random であるためには,

$$\exists c \in \mathbb{N}\ \forall n \in \mathbb{N}\ K(X \upharpoonright n) \geq n - c$$

が成り立つことが必要十分条件である (Schnorr [Sch71b]).

X が Martin-Löf random であるとき, 極限値 (1.1) は 1/2 であり, かつ, X は P-bi-immune である. ここで,「X が P-bi-immune である」とは,「X の無限部分集合で多項式時間計算可能なものが存在せず, かつ, X と共通部分をもたない無限部分集合で多項式時間計算可能なものも存在しない」ことをいう [BDG90, SY04]. 以上により,「不規則なオラクル」を Martin-Löf random なオラクルとして定式化すると, 要請 1 および要請 3 がみたされないことがわかる. 同様に,「不規則なオラクル」をランダム・オラクルとして定式化すると, 要請 1 および要請 3 がみたされない.

本稿では, 以下の定理を示す.

定理 (定理 1) 任意の Martin-Löf random なオラクルは, 性質 1 をもつ.

上記の定理について詳しくは第 6 節において論じる.

Resource bounded randomness and resource bounded genericity: Schnorr [Sch71a, Sch71b] は Martin-Löf randomness の概念に不満をもち, resource bounded measure と resource bounded randomness の概念を提唱した. (Martingale に付随する戦略アルゴリズムの) 計算時間を制限する関数 $t(n)$ が与えられると, それに応じて $t(n)$-measure の概念と $t(n)$-randomness の概念が定まる. Lutz [Lu92] はこれらの概念を再発見し, 発展させた. $t(n)$-randomness の概念は, Ambos-Spies らによる $t(n)$-genericity の概念 [AFH88, Am96] と関係が深い.

$t(n)$ が $\forall n\ t(n) \geq n^2$ という性質をもつ関数で, X がオラクルであるとき, 以下が成り立つ.

- [ANT96, ATZ97] If X is $t(n)$-random then X is $t(n)$-generic.
- [ANT96] If X is $t(n)$-generic then X is DTIME($t(2^{n-1})$)-bi-immune.

ここで,「X が DTIME($t(2^{n-1})$)-bi-immune である」とは,「X の無限部分集合で $O(t(2^{n-1}))$ 時間計算可能なものが存在せず, かつ, X と共通部分をもたない無限部分集合で $O(t(2^{n-1}))$ 時間計算可能なものも存在しない」ことをいう. したがって,「不規則なオラクル」を「任意の多項式 p に対して p-generic なオラクル」として定式化すると, 要請 3 がみたされないことがわかる.「不規則なオラクル」を「任意の多項式 p に対して p-random なオラクル」として定式化する場合も同様であ

る．Resource-bounded randomness および genericity についての総合報告としては [AM97] を，resource-bounded genericity についての総合報告としては [Am96] を参照されたい．

Time complexiy, circuit complexity and their variants: オラクル X の時間計算量が高くても X が $\forall n \in \mathbb{N}\, X(2n) = X(2n+1)$ という規則性を持つことは可能である．したがって，「不規則なオラクル」を「時間計算量がじゅうぶん高いオラクル」として定式化すると，要請 2 の 2 がみたされないことがわかる．「不規則なオラクル」を「回路計算量がじゅうぶん高いオラクル」として定式化する場合も同様である．

Min-entropy and Nisan-Zuckerman-type extractor: 放射性元素，空気の乱流，半導体の熱学的ノイズなどの物理的乱数源からの乱数抽出の歴史は長い．Zuckerman [Zuc90] はこの種の乱数源（weak random source）に対する汎用の数学的モデル作りを試み，確率分布のランダム性の程度を測るために min-entropy に注目した．Shannon のエントロピーが一種の平均量であるのに対して，Nisan-Zuckerman の流儀では，エントロピーの最小量に着目する．後に，Nisan は Zuckerman とともに，min-entropy に基づいて乱数抽出器（extractor）の概念を定義した [NZ93, NZ96]．

定義 3 1. [Zuc90] Suppose that X is a distribution on $\{0,1\}^n$. The *min-entropy of* X (denoted by $H_\infty(X)$) is defined as follows.

$$H_\infty(X) = \min_{x \in \{0,1\}^n} \log \frac{1}{\mathrm{Prob}[X = x]}.$$

2. [Sh02] Suppose that ε is a positive number. Two distributions P, Q over the same domain T are *ε-close* if it holds that

$$\frac{1}{2} \cdot \sum_{x \in T} |P(x) - Q(x)| \leq \varepsilon.$$

3. [NZ93, NZ96] Suppose that k is a positive integer and ε is a positive number. A function

$$\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

is called a *(k, ε)-extractor* if for every distribution X on $\{0,1\}^n$ with $H_\infty(X) \geq k$, the distribution $\mathrm{Ext}(X, U_d)$ (where U_d is uniformly distributed in $\{0,1\}^d$) is ε-close to the uniform distribution on $\{0,1\}^d$.

定義 3 の項目 3 において，関数 Ext の第 1 引数は min-entropy の高い乱数源からの標本である．第 2 引数は真にランダムな変数と仮定されており，乱数の種とよばれる．

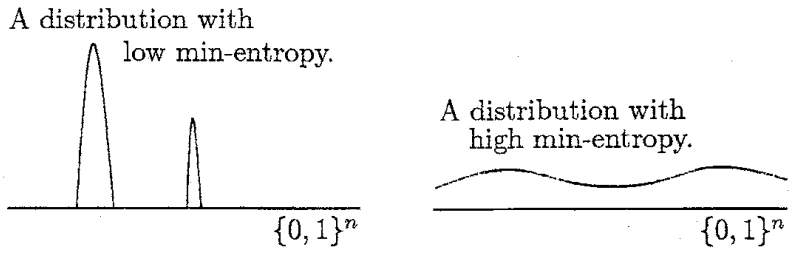

図 2: Min-entropy

定義 3 の min-entropy および extractor の概念を我々の問題設定に活用できるかどうか，本稿の筆者らは知らない．本稿では乱数源として手描き曲線のみを用い，それ以外に真にランダムな変数の入力を要求しない．この意味において，Nisan-Zuckerman 流のアプローチは我々の課題に（少なくとも，直接的には）応用できない．

定義 3 の extractor およびその変種についての総合報告としては [NTs99, Sh02] を参照されたい．

Pseudorandom number generator in Goldreich's book: 1980 年代以降，計算量理論と暗号研究の境界分野においては，特定の計算資源のもとで一様分布と区別不可能な分布を擬似乱数とみなす研究も行なわれている [Go99, Chapter 3].「不規則なオラクル」を「特定の計算資源のもとで一様分布と区別不可能なオラクル」として定式化すると，要請 1 がみたされない．

1.5 鈴木による修正版アルゴリズムの概要

鈴木による修正版アルゴリズムは，次々にビットマップファイルを受け取りながら，概ね以下の作業を行う．最初に正の整数 C_1, C_2 を固定し，$C_3 = \mathrm{floor}(\log_2(C_1 + C_2))$ とおく．ただし，$\mathrm{floor}(x)$ は x を超えない最大の整数を表す．本稿では $C_1 = 60$, $C_2 = 2$ とするので $C_3 = 5$ である．C_1, C_2 の値は，与えられるビットマップファイルのピクセル数（本稿では 256×256 ピクセルに固定）に依存してもよいが，個々のビットマップファイルには依存しないものとする．

STEP1: 川西のアルゴリズムと同様にして，以下のビット列 a を得る．

$$a = a_{0,0} \cdots a_{0,N-1} a_{1,0} \cdots a_{1,N-1} \cdots a_{N-1,0} \cdots a_{N-1,N-1}$$

STEP2: STEP1 で得たビット列を C_1 ビットずつに区切る．

それを $a^{(0)} = a_0^{(0)} \cdots a_{C_1-1}^{(0)}$, $a^{(1)} = a_0^{(1)} \cdots a_{C_1-1}^{(1)}$, $\cdots$ とする．

$a^{(0)}$ において，各々の連の長さに C_2 を加えた数を 2 進数で表し，それらの 2 進数を連接させて得られるビット列の最初の C_3 桁として定まるビット列を $b^{(0)}$

とする．以下，$a^{(1)}, a^{(2)}, \cdots$ に対して同様の操作を行ってビット列 $b^{(1)}, b^{(2)}, \cdots$ を作る．そして，$b^{(0)}, b^{(1)}, b^{(2)}, \cdots$ をすべて連接させて新たなビット列を作る．

STEP3: STEP2で得たビット列を $b_0 \cdots b_{2n-1+k}$ とする．ただし，n は自然数で，k は0または1である．ここで，

$$b_0 \text{ xor } b_{2n-1+k}, b_1 \text{ xor } b_{2n-2+k}, \cdots, b_j \text{ xor } b_{2n-j-1+k}, \cdots, b_{n-1} \text{ xor } b_{n+k}$$

を次々に出力していく．

そして，次の行列を受け取る．以下，繰り返し．

鈴木による修正版アルゴリズムのより正確な記述は第4節のアルゴリズム3を参照のこと．また，検定の結果については，第3節で述べる．第6節では，以下の定理を示す．

定理（定理8） 上記の修正版アルゴリズムは，性質2をもつ写像（汎関数）を与える．

より詳しくは以下のとおりである．与えられたオラクル A に対し，ビット列 $A(z(0))A(z(1))\cdots A(z(N^2-1))$（ただし，短さ優先の辞書式順序で $(k+1)$ 番目のビット列を $z(k)$ で表す），を修正版アルゴリズムのSTEP2, 3で処理してビット列 $B(z(0))B(z(1))\cdots B(z(M-1))$ を作り（M は定数），次に $A(z(N^2))A(z(N^2+1))\cdots A(z(2N^2-1))$ を同様に処理して $B(z(M))B(z(M+1))\cdots B(z(2M-1))$ を作り，$\cdots$ という操作を繰り返してオラクル B を作る．このとき，A に B を対応させる写像（汎関数）は，性質2をもつ．

2 用語と記法

自然数 x に対し，x を超えない最大の整数を $\mathrm{floor}(x)$ で表す．以下，本節では the relativized propositional calculus および Dowd オラクルについて説明する．説明の中で，講演 [鈴木99] のアブストラクトの一部を編集の上，流用する．

ビット列全体の集合を $\{0,1\}^*$ で表す．オラクルをその特性関数と同一視して $\{0,1\}^*$ から $\{0,1\}$ への関数をオラクルと言う．また，あるオラクルの定義域を，ビット列のある有限集合に制限して得られる関数を，**強制条件** (forcing condition) という．n 個 $(n \geq 1)$ の命題変数に対する演算子として ξ^n というクエリー記号を導入する．$\xi^n(q_1, \ldots, q_n)$ の解釈は，おおまかに言えば「ビット列 $q_1 \cdots q_n$ がオラクルに属する」ということである．The relativized propositional calculus (RPC) は，通常の命題論理にクエリー記号の集合 $\{\xi^n : n \in \mathbb{N}\}$ を加えたものである．クエリー記号 ξ^n の解釈をもう少し詳しく述べよう．各オラクル A に対し，n-変数ブール関数 A^n を導入

する．話を簡単にするため，ここでは A^3 の意味を説明する．まず，長さ 3 のビット列全体の集合 $\{0,1\}^3$ と $\{0,1\}^*$ に，(短さ優先の) 辞書式順序を導入する．ここで λ は長さゼロのビット列を表す．

$$\begin{array}{ll} \{0,1\}^3: & 000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111. \\ \{0,1\}^*: & \lambda,\ 0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ \ldots \end{array}$$

いま，$\{0,1\}^*$ の最初の 8 個の元の集合 $\{\lambda,\ldots,000\}$ に注目する．以下の図式が可換となるように関数 A^3 を定める．但し，$\simeq$ は辞書式順序に関する順序同型である．

$$\begin{array}{ccc} & & A^3 \\ \{0,1\}^3 & \longrightarrow & \{0,1\} \\ \simeq \big\downarrow & \nearrow & A \restriction \{\lambda,\ldots,000\} \\ \{\lambda,\ldots,000\} & & \end{array}$$

オラクル A が与えられたとき，$\xi^3(q_1,q_2,q_3)$ を $A^3(q_1q_2q_3)$ として解釈する．この一見遠回りな定義をする理由は，(1) こう定義すると，$\xi^n(q_1,\ldots,q_n)=\xi^{n+1}(0,q_1,\ldots,q_n)$ となって ξ^n の情報が ξ^{n+1} に引き継がれること，(2) 命題論理の演算子の引数は，ある特定の値でなければならないこと，の 2 点にある．

短さ優先の辞書式順序で $(k+1)$ 番目のビット列を $z(k)$ で表す．たとえば，$\lambda=z(0)$ となる．ここで自然数 k の n ビット 2 進表記と k 自身とを同一視すれば，上記の A^n の定義を以下のごとく簡潔に言い表すことができる．もっとも，これは少々粗雑な記法であるが．

$$A^n(k)=A(z(k)).$$

通常の恒真式全体の集合を TAUT で表し，オラクル A に関する恒真式全体の集合を TAUT^A で表す．また，クエリー記号の出現をちょうどひとつ持つ式を one-query formula と呼ぶ．One-query formula であって，なおかつ TAUT^A の元になっているものは A に関する one-query tautology と呼ばれる．A に関する one-query tautology 全体の集合を $1\mathrm{TAUT}^A$ で表す．同様にして，自然数 r に対して r-query formula, オラクル A に関する r-query tautology, $r\mathrm{TAUT}^A$ の概念を導入する．F がクエリー記号付き命題論理式で，かつ，強制条件 S の拡張になっている任意のオラクル X に対して F がトートロジーになるとき，「S は F を**強制する**」という．オラクル A が t ジェネリック・オラクルであるとは，ある多項式 p があって，任意の $F\in\mathrm{TAUT}^A$ に対し，A の (特性関数の) 有限部分 S があって，以下の条件が成り立つことを言う：「S は F が恒真式であることを強制し，かつ，S の (定義域の) サイズは高々 $p(|F|)$ である」．

自然数 r に対し，オラクル A が r-Dowd オラクル（Dowd の意味での r-generic oracle）であるとは，t ジェネリック・オラクルの定義において TAUT^A を $r\mathrm{TAUT}^A$

に置きかえた主張がなりたつことをいう．任意の正の整数rに対し，r-Dowd オラクル全体のクラスはカントル空間において測度1となる（証明のあらましは[Do92]，厳密で詳細な証明は[Su01, Su02]を参照されたい）．ただし，tジェネリックオラクルは存在しない[Su01]．RPC および Dowd オラクルについて，より詳しくは[Do92, Su01, Su02, Su05]を参照されたい．

Martin-Löf randomness については第1節の定義2および Calude の本[Ca02]を，計算量理論の基礎事項については[BDG88, BDG90]を，確率論については[Fe68, Fe71]を，統計の基礎事項については[石井95]を，乱数の基礎事項については[Ge03, Kn98, 伏見89, 脇本70]をそれぞれ，参照されたい．また，BMP 形式の画像ファイルについては[宮坂04]が，そして広く用いられている擬似乱数生成プログラムの問題点については[PM88, 和田04]が参考になるであろう．

3 川西の実験

本節では川西のアルゴリズムと連の検定について述べる．

3.1 連の検定について

連の検定とは，擬似乱数の数字の並び方の無規則性を検証する検定法のひとつであり，その特徴として「連」の数に注目している．

「連」とは，本稿の場合，00000 のような同じ数字が続くひとかたまりのことをいう（例えば，010011100 なら連の数は5）．連の数が（期待値に比べて）非常に多い列においては「0がきたら次は1だろう」という予測ができるため，インフォーマルに言えば，そのような列はランダム性の度合いが低い．逆に連の数が非常に少ない列においては「0がきたら次も0だろう」という予測ができるため，そのような列もやはりランダム性の度合いが低い．真にランダムなビット列における連の個数の分布は，一定の条件の下で，近似的に正規分布に従う．本稿では，以下の統計量Vを用いる．

統計量Vの説明： 自然数n_0, n_1を固定し，文字「0」の書かれた札n_0枚と文字「1」が書かれた札n_1枚を一列に並べる試行を行なうとする．自然数n_0, n_1が十分大きく，文字の並べ方が（真に）ランダムであるとき，これらの試行を多数繰り返すと，連の個数の分布は近似的に正規分布に従う．$n = n_0 + n_1$とおき，$b = n_1/n$とおくと連の個数の平均値はおよそ$2nb(1-b)$であり，分散は$4nb^2(1-b)^2$である[脇本70]．ここでnは列の長さ，bは1（黒）の割合を表す．このとき，

$$\frac{(\text{連の個数}) - 2nb(1-b)}{2b(1-b)\sqrt{n}} \tag{3.1}$$

は近似的に標準正規分布に従う（注，標準正規分布は平均0，分散1であり，95%の確率で-1.96 から 1.96 の間の値をとる）．そこで，(3.1) における連の個数，n，およ

びbのそれぞれを，アルゴリズムが出力した列（標本）の連の個数，その列の長さ，その列における1（黒）の割合で置きかえた統計量を（その標本の）Vと定める．

本稿では，与えられたビットマップファイルひとつをあるアルゴリズムで処理して得たビット列に対して統計量Vを求め，その絶対値が1.96以下であるかどうか調べる．これが，もっとも素朴な意味での連の検定である．また，アルゴリズムを固定し，ファイルの集合（50枚，あるいは100枚）に対して統計量Vの平均や分散を求めたり，適合度検定を行う．

3.2 BMPファイルを行列に変換する

ウィンドウズ®のペイントなど,ペイントソフトで描かれた曲線のビットマップ・ファイル(画像形式は24ビットBMPで，キャンバスのサイズは256×256ピクセル)を行列に変換する．ビットマップの上からi番目,左からj番目のピクセルがキャンバスと同じ色であるとき行列の要素$a_{i,j}$を0と定め,そうでないとき1と定める．以下，色を表す数としての0を「白」，1を「黒」ということもある．

3.3 一定以上の長さをもつ連を圧縮する（失敗例）

手描き曲線のBMPファイルをそのまま行列化したものは，統計学的にみて不自然に長い連をもち，擬似乱数としての使用に適さない．

そこで川西は，長さ10以上の連に対し，10で割った余り分に圧縮するという方法を試した．例えば00000000000(長さ11の連）は0(長さ1の連）に圧縮される．

この方法の限界を示す経験的事実

- 圧倒的に白が多いBMPファイルにこの圧縮を施してみたところ，1の割合は0.5と比べて著しく低かった（あるファイルでは0.25）．
- 0と1の割合がほぼ同じの絵で，統計量Vの絶対値が1.96より大なるもの（素朴な意味で，連の検定に合格しないもの）にこの圧縮を施してみたところ，依然として統計量Vの絶対値が1.96より大なるままにとどまることが多かった．統計量Vについては第3.1小節を参照されたい．

3.4 連の長さを2進数化する

川西は,次に連を2進数化するアルゴリズムを考案した．このアルゴリズムは，ビットマップファイルを次々に受け取り，以下の作業を行なう．

アルゴリズム 1 （川西のアルゴリズム 試作版）

STEP1: ビットマップファイルを行列 (2 次元配列) に変換する．さらに，行列を 1 次元配列に変換する．

- 行列の一番上の行の一番左の成分から順番に，英文を読む順序で 1 ビットずつ，一次元配列 la に格納する (la は，linear array の略)．

STEP2: 連の長さを 2 進数化していく．

例として配列 la が 1100011100000 である場合について述べる．

- 連の長さは 2,3,3,5 である．これらを 2 進数にすると，それぞれ 1,11,11,101 となる（以下の注意 2 を参照）．

注意 2 連の長さが 1,2 の場合は例外扱いする．まず，連の長さが 2 の場合は,2 進数「1」を対応させる．連の長さが 1 の場合は，配列 la の番地（配列の左から i 番目を i 番地とする) によって区別し，番地が偶数であれば 0，奇数であれば 1 を対応させる．

- それぞれの 2 進数の一番大きい位，つまり最上位ビットの 1 を削除する．その結果，1,1,1,01 を得る．ただし，連の長さが 1 の場合は，この操作を行わない．
- それぞれを左右逆転（ABC→CBA）し，一次元配列 be に格納する (be は，binary encoding の略)．このとき，配列 be : 1 1 1 1 0

STEP3: STEP2 の配列 be の各ビットをひとつずつ出力していく．

そして，次のビットマップファイルを受け取る．以下，繰り返し．

アルゴリズム 1，終わり．

連の検定による配列 be の検定結果

川西は，自身がペイントツールで描いた落書きビットマップ 200 枚を用いて，実験を行った．この実験により得られた 200 個の配列 be の大部分において,1 の割合は 0.5（つまり 50 %）前後であったが，統計量 V の絶対値は 1.96 より大きかった（連の数が期待値よりはるかに多かった）．

3.5 XOR を用いる

川西はアルゴリズム 1 にさまざまなビットマップファイルを与えてその出力を調べてみたが，統計量 V の絶対値が 1.96 より大なるもの（素朴な意味で，連の検定に不合格であるもの）が多かった．しかし 1 の割合は 0.5 前後のものが多かった．

そこで,アルゴリズム 1 の出力（配列 be）を XOR（排他的論理和）を用いて加工し,新たな配列を作ることにした. 新たなアルゴリズムは，ビットマップファイルを次々に受け取りながら以下の作業を行う.

アルゴリズム 2 （川西のアルゴリズム）

STEP1: アルゴリズム 1 の STEP1 と同じ.

STEP2: アルゴリズム 1 の STEP2 と同じ.

STEP3: 各 $i(1 \leq i \leq (1/2)$(配列 be の長さ)) に対し，配列 be の先頭から i 番目（i は正の整数）の要素と,配列 be の後ろから i 番目の要素の排他的論理和 (xor) をとり,新しい配列 myXOR の i 番目に格納する. 配列 myXOR の各ビットをひとつずつ出力していく.

そして，次のビットマップファイルを受け取る. 以下，繰り返し.

アルゴリズム 2，終わり.

連の検定による配列 myXOR の検定結果

さまざまなビットマップファイルをアルゴリズム 2 に与え，配列 myXOR を作ったところ，その多くにおいて 1 の割合は 0.5 前後であり，統計量 V の絶対値が 1.96 以下であった. 以下に,アルゴリズムの出力についての実験結果を示す. 参考のためにアルゴリズム 3 （鈴木による修正版アルゴリズム，第 4 節を参照のこと）についての結果も併せて載せる.

実験データと結果

表 2 は test1.bmp–test100.bmp という 100 枚のビットマップファイルを 2 種類のアルゴリズム（アルゴリズム 2 と 3）に与えたときに，それぞれのアルゴリズムが出力したビット列についての実験結果をおよその値で示したものである. これら 100 枚のビットマップは,川西がマウスで描いたものである. 意味のない曲線を描いたファイルと，人物の顔や文字などを漫画風に描いたファイルが混合したものである. マウスは筆圧に対応しておらず，マウスで描いた線の太さは一定である. 画像はすべてモノクロであるが，ビットマップファイルのファイル形式には 24 ビット BMP 形式（多くの色を表示できる画像ファイル形式の一種）を用いた.

表の「χ^2 検定（標本数 M，自由度 $2n+1$）」という項目について説明する. 統計量 V の標本 M 個を用意し，標準正規分布に対する適合度のカイ 2 乗検定を行う. 正の実数 $x_1, x_2, \cdots, x_n$（ただし $x_1 < x_2 < \cdots < x_n$）をうまく選び，$2n+2$ 個の事象「$E_1 : x \leq -x_n$」,「$E_2 : -x_n < x \leq -x_{n-1}$」, $\cdots$,「$E_{n+1} : -x_1 < x \leq 0$」,「$E_{n+2} : 0 < x \leq x_1$」, $\cdots$,「$E_{2n+1} : x_{n-1} < x \leq x_n$」,「$E_{2n+2} : x_n < x$」がいず

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.494	0.494
V の平均値	-0.193	-0.187
χ^2 検定 (標本数 100，自由度 19)	合格	合格
α	0.0760	0.0518

表 2: test1.bmp–test100.bmp

れも，標準正規分布において確率 $1/(2n+2)$ をもつようにする．ここで，帰無仮説「統計量 V の標本 M 個は上記 $2n+2$ 個の区間の各々に $M/(2n+2)$ 個ずつ分布している」を有意水準 5 パーセントで片側検定する．各 $i = 1, 2, \cdots, 2n+2$ に対し，事象 E_i の観測度数を f_i とする．このとき，自由度 $2n+1$ のカイ 2 乗分布において，

$$\sum_{i=1}^{2n+2} \frac{(f_i - M/(2n+2))^2}{M/(2n+2)} \tag{3.2}$$

よりも大きい値が得られる確率を α であるとする．ここで，$\alpha < 0.05$ のとき，帰無仮説は棄却される．そのとき，本稿の表には「不合格」と記す．$\alpha \geq 0.05$ のとき，帰無仮説は棄却されない．そのとき，本稿の表には「合格」と記す．

次に，表 3 は test101.bmp–test200.bmp という 100 枚のビットマップファイルについての実験結果である．これら 100 枚は，川西がペンタブレット (WACOM 社製 intuos 3® PTZ-630，筆圧対応) で描いた落書きである．意味のない曲線を描いたファイルと，人物の顔や文字などを漫画風に描いたファイルが混合したものである．

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.497	0.496
V の平均値	0.139	-0.117
χ^2 検定 (標本数 100，自由度 19)	不合格	合格
α	0.00528	0.284

表 3: test101.bmp–test200.bmp

256×256 ピクセルの画像を入力として与えたとき，アルゴリズム 3 が出力するビット列の長さは常に一定しており，2730 である．一方，アルゴリズム 2 が出力するビット列の長さは一定しておらず，test1.bmp–test100.bmp の 100 枚の画像に対する平均値はおよそ 12,000 である．

後ほど，第 5 節で，より詳しい実験結果を示す．

4 修正版アルゴリズム

4.1 修正版アルゴリズムの定義

鈴木による修正版アルゴリズムは，ビットマップファイルを次々に受け取りながら以下の作業を行う．

アルゴリズム 3 （鈴木による修正版アルゴリズム）

最初に正の整数C_1, C_2を固定し，$C_3 = \mathrm{floor}(\log_2(C_1+C_2))$とおく．ただし，$\mathrm{floor}(x)$は$x$を超えない最大の整数を表す．本稿では$C_1 = 60$, $C_2 = 2$とするので$C_3 = 5$である．C_1, C_2の値は，与えられるビットマップファイルのピクセル数（本稿では256×256ピクセルに固定）に依存してもよいが，個々のビットマップファイルには依存しないものとする．

STEP1: 川西のアルゴリズムと同様にして，以下のビット列aを得る．

$$a = a_{0,0} \cdots a_{0,N-1} \cdots a_{N-1,0} \cdots a_{N-1,N-1}$$

STEP2: STEP1で得たビット列をC_1ビットずつに区切る．

それを$a^{(0)} = a_0^{(0)} \cdots a_{C_1-1}^{(0)}$, $a^{(1)} = a_0^{(1)} \cdots a_{C_1-1}^{(1)}$, $\cdots$とする．

$a^{(0)}$において，各々の連の長さにC_2を加えた数を2進数で表したものを$r_0^{(0)}$, $r_1^{(0)}, \cdots, r_j^{(0)}, \cdots$とする．各$j$に対し，$r_j^{(0)}$から最上位ビットの1を取り除き，左右逆転させたものを$s_j^{(0)}$とする．$s_0^{(0)}, s_1^{(0)}, \cdots, s_j^{(0)}, \cdots$を連接させて得られるビット列の最初の$C_3$桁からなるビット列を$b^{(0)}$とする．以下，$a^{(1)}, a^{(2)}, \cdots$に対して同様の操作を行ってビット列$b^{(1)}, b^{(2)}, \cdots$を作る．そして，$b^{(0)}, b^{(1)}, b^{(2)}, \cdots$をすべて連接させて新たなビット列$b$を作る（図3）．

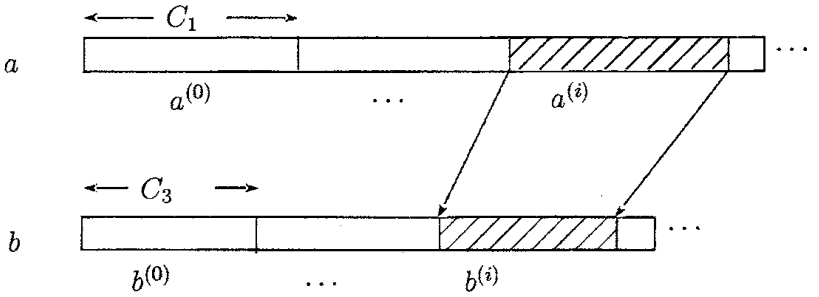

図 3: ブロックごとに処理する

STEP3: STEP2で得たビット列を$b_0 \cdots b_{2n-1+k}$とする．ただし，nは自然数で，kは0または1である．ここで，

$$b_0 \,\mathrm{xor}\, b_{2n-1+k}, b_1 \,\mathrm{xor}\, b_{2n-2+k}, \cdots, b_{n-1} \,\mathrm{xor}\, b_{n+k}$$

を次々に出力していく.

そして，次のビットマップファイルを受け取る．以下，繰り返し.

アルゴリズム 3，終わり.

4.2 修正版アルゴリズムを用いた労働の効率

上記のアルゴリズムを使って乱数表を作るためには，人間が次々にビットマップファイルを描く必要がある．ここでは話を単純化して，労働量はビットマップのキャンバスの総ピクセル数によって表されると考える．このとき，一辺 N ピクセルの正方形のキャンバスをもつビットマップ M 枚を描く労働量は N^2M となる．一辺 N ピクセルの正方形のキャンバスをもつビットマップ M 枚を上記のアルゴリズムに与えたとき，出力されるビット列の桁数は以下によって与えられる.

$$(1/2)C_3\,\mathrm{floor}(\frac{N^2}{C_1})M \fallingdotseq \frac{(\log_2(C_1+C_2))N^2M}{2C_1}$$

$N=256$, $C_1=60$, $C_2=2$ の場合，上記の式のおよその値は

$$\frac{1}{2}\cdot\frac{5}{60}256^2M=\frac{1}{24}256^2M \fallingdotseq 0.0417\times 256^2M$$

となる．これが労働量 256^2M によって出力されるビット列の桁数であり，それは労働量の一次関数である.

C_3 を $\log_2(C_1+C_2)$ と定めることに対して「C_3 が小さすぎる」という批判があり得るが，この批判が的を得ているのは労働量 256^2M が小さい場合（C_1, C_2 に比べてあまり大きくない場合）に限られる．出力されるビット列の桁数が労働量の対数関数になっているわけではないことに注意されたい.

5 詳細実験

本節では 500 枚のビットマップファイル test201.bmp–test700.bmp を用い，絵の種類（無意味な曲線と漫画風の絵，マウスで描いた絵とペンタブレットで描いた絵）ごとに，川西のアルゴリズムと修正版アルゴリズムの出力を比較した実験結果を示す．いずれのファイルも，およそ 1 分で描いたものである．統計量 V の定義については第 3.1 小節を，「χ^2 検定（標本数 M，自由度 $2n+1$）」という項目の説明は第 3.5 小節をそれぞれ参照されたい.

5.1 マウスで描いた無意味な曲線

100枚のビットマップファイルtest201.bmp-test300.bmpはいずれも，マウスで描いた無意味な曲線である．test201〜250.bmpは川西が，test251〜300.bmpは鈴木が描いた．

表4はtest201〜300.bmpをアルゴリズム2およびアルゴリズム3で変換した結果を示す．test201〜250.bmpの組とtest251〜300.bmpの組に分けて実験した結果は表5および表6のとおりである．

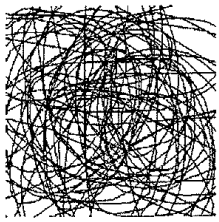

図 4: test201.bmp

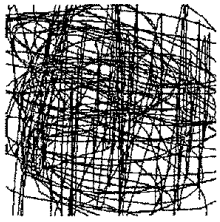

図 5: test229.bmp

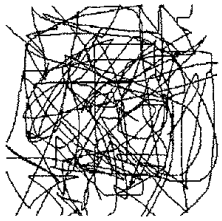

図 6: test266.bmp

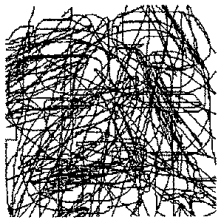

図 7: test288.bmp

5.2 ペンタブレットで描いた無意味な曲線

100枚のビットマップファイルtest301.bmp-test400.bmpはいずれも，ペンタブレット（筆圧対応）で描いた意味のない曲線である．test301〜350.bmpは川西が，test351〜400.bmpは鈴木が描いた．

表7はtest301〜400.bmpをアルゴリズム2およびアルゴリズム3で変換した結果を示す．test301〜350.bmpの組とtest351〜400.bmpの組に分けて実験した結果は表8および表9のとおりである．

図 8: test311.bmp

図 9: test342.bmp

図 10: test354.bmp

図 11: test397.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.497	0.499
1 の割合の分散	3.90×10^{-5}	7.85×10^{-5}
V の平均値	-0.295	0.0442
V の分散	0.995	1.0718
出力の長さの平均値	12562	2730（一定）
出力の長さの分散	3.267×10^{6}	0
χ^2 検定 （標本数 100，自由度 19） α	合格 0.0629	合格 0.902

表 4: test201.bmp–test300.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.495	0.500
V の平均値	-0.303	-0.0430
V の分散	0.828	0.918
χ^2 検定（標本数 50，自由度 9） α	不合格 0.00888	合格 0.956

表 5: test201.bmp–test250.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.499	0.499
V の平均値	-0.287	0.131
V の分散	1.163	1.211
χ^2 検定（標本数 50，自由度 9） α	合格 0.658	合格 0.699

表 6: test251.bmp–test300.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.495	0.497
1 の割合の分散	5.34×10^{-5}	8.93×10^{-5}
V の平均値	-0.101	0.0896
V の分散	0.963	0.967
出力の長さの平均値	5951	2730（一定）
出力の長さの分散	2.34×10^{6}	0
χ^2 検定 （標本数 100，自由度 19） α	合格 0.735	合格 0.305

表 7: test301.bmp–test400.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.495	0.496
V の平均値	-0.0824	0.225
V の分散	1.259	0.915
χ^2 検定（標本数 50，自由度 9） α	合格 0.740	合格 0.494

表 8: test301.bmp–test350.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.495	0.498
V の平均値	-0.120	-0.0455
V の分散	0.666	0.982
χ^2 検定（標本数 50，自由度 9） α	合格 0.122	合格 0.956

表 9: test351.bmp–test400.bmp

5.3 マウスで描いた漫画風の絵

100 枚のビットマップファイル test401.bmp–test500.bmp はいずれも，マウスで描いた絵であり，そのほとんどは人物，動物あるいは怪物の顔を漫画風に描いたものである．test401–450.bmp は川西が，test451–500.bmp は鈴木が描いた．

表 10 は test401–500.bmp をアルゴリズム 2 およびアルゴリズム 3 で変換した結果を示す．test401–450.bmp の組と test451–500.bmp の組に分けて実験した結果は表 11 および表 12 のとおりである．

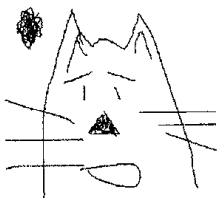

図 12: test426.bmp

図 13: test431.bmp

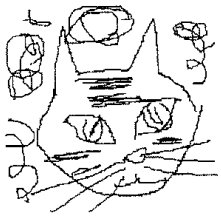

図 14: test456.bmp

図 15: test499.bmp

5.4 ペンタブレットで描いた漫画風の絵

100 枚のビットマップファイル test501.bmp–test600.bmp はいずれも，ペンタブレット（筆圧対応）で描いた絵であり，そのほとんどは人物，動物あるいは怪物の顔を漫画風に描いたものである．test501–550.bmp は川西が，test551–600.bmp は鈴木が描いた．

表 13 は test501–600.bmp をアルゴリズム 2 およびアルゴリズム 3 で変換した結果を示す．test501–550.bmp の組と test551–600.bmp の組に分けて実験した結果は表 14 および表 15 のとおりである．

図 16: test509.bmp

図 17: test514.bmp

図 18: test595.bmp

図 19: test600.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.500	0.478
1 の割合の分散	7.22×10^{-5}	2.25×10^{-4}
V の平均値	-0.112	-0.0925
V の分散	1.067	1.488
出力の長さの平均値	4423	2730（一定）
出力の長さの分散	2.69×10^{6}	0
χ^2 検定 （標本数 100，自由度 19） α	合格 0.603	合格 0.522

表 10: test401.bmp–test500.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.503	0.473
V の平均値	-0.0825	-0.375
V の分散	1.091	1.816
χ^2 検定（標本数 50，自由度 9）	合格	合格
α	0.851	0.122

表 11: test401.bmp–test450.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.497	0.483
V の平均値	-0.141	0.190
V の分散	1.041	1.001
χ^2 検定 （標本数 50，自由度 9）	合格	合格
α	0.817	0.384

表 12: test451.bmp–test500.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.496	0.496
1 の割合の分散	5.04×10^{-5}	1.29×10^{-4}
V の平均値	-0.0507	0.0889
V の分散	1.090	1.004
出力の長さの平均値	4847	2730（一定）
出力の長さの分散	1.20×10^{6}	0
χ^2 検定 （標本数 100，自由度 19） α	合格 0.849	合格 0.657

表 13: test501.bmp–test600.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.497	0.495
V の平均値	-0.0524	0.0525
V の分散	0.896	1.095
χ^2 検定（標本数 50，自由度 9） α	合格 0.911	合格 0.911

表 14: test501.bmp–test550.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.496	0.498
V の平均値	-0.0490	0.125
V の分散	1.284	0.911
χ^2 検定（標本数 50，自由度 9） α	合格 0.419	合格 0.384

表 15: test551.bmp–test600.bmp

5.5 マウスで描いた幾何学的デザイン

100 枚のビットマップファイル test601.bmp–test700.bmp はいずれもマウスで描いたもので，円，楕円，直線などから構成されたデザインである．test601–650.bmp は川西が，test651–700.bmp は鈴木が描いた．

表 16 は test601–700.bmp をアルゴリズム 2 およびアルゴリズム 3 で変換した結果を示す．test601–650.bmp の組と test651–700.bmp の組に分けて実験した結果は表 17 および表 18 のとおりである．

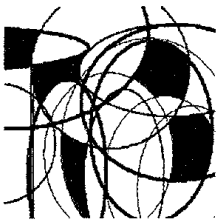

図 20: test647.bmp

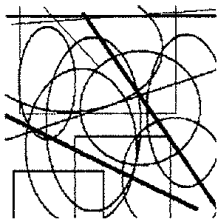

図 21: test650.bmp

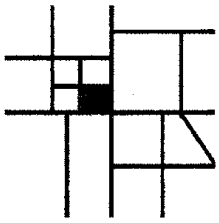

図 22: test687.bmp

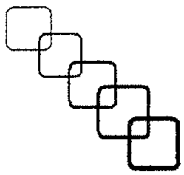

図 23: test700.bmp

6 乱数抽出の意味論

この節では，我々の意味論を紹介する．

不規則な対象のもつべき性質　オラクル D が

性質 1「任意の自然数 r に対して，D は r-Dowd オラクル（Dowd の意味での r ジェネリック・オラクル）である」

という性質をもつとき，我々は「D が不規則である」と考える．

定理 1 *Suppose that an oracle A is Martin-Löf random.*
Then A is r-Dowd for any positive integer r.

不規則性を保つ写像のもつべき性質　f はオラクル全体のクラスからオラクル全体のクラスへの写像であり，かつ，以下の性質 0 をもつとする．

性質 0「多項式時間タイマー付き（polynomial-time-clocked）オラクル・チューリングマシン $M^{\sim}$ が存在して，任意のオラクル A と任意のビット列 u に対して $M^A(u) = f(A)(u)$ が成り立つ」

このような f がさらに以下の性質 2 をもつとき，我々は「f が不規則性を保つ」と考える．

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.496	0.468
1 の割合の分散	0.00198	0.00112
V の平均値	0.0305	-0.792
V の分散	8.540	5.286
出力の長さの平均値	2895	2730 (一定)
出力の長さの分散	2.17×10^6	0
χ^2 検定 (標本数 100，自由度 19) α	不合格 1.96×10^{-13}	不合格 8.88×10^{-20}

表 16: test601.bmp–test700.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.492	0.489
V の平均値	-0.286	0.0700
V の分散	1.827	2.383
χ^2 検定（標本数 50，自由度 9） α	合格 0.350	合格 0.350

表 17: test601.bmp–test650.bmp

	アルゴリズム 2	アルゴリズム 3
1 の割合の平均値	0.500	0.446
V の平均値	0.347	-1.653
V の分散	15.054	6.704
χ^2 検定（標本数 50，自由度 9） α	不合格 7.85×10^{-9}	不合格 4.31×10^{-22}

表 18: test651.bmp–test700.bmp

性質 2 「性質 1 をもつ任意のオラクル D に対して，$f(D)$ は性質 1 をもつ」

オラクル全体のクラスからオラクル全体のクラスへの写像 f が性質 0 をもち，かつ，

性質 2′ 「任意の自然数 r に対して自然数 s が存在して，任意の s-Dowd オラクル D に対して，$f(D)$ は r-Dowd オラクルである」

という性質をもてば，容易にわかるように，f は性質 2 をもつ．

また，性質 2 をもつ写像は合成に関して閉じている．

命題 2 *[Su02] Suppose that D is a 1-Dowd oracle. Then, D does not have polynomial size circuit ($D \notin \mathrm{P/poly}$). In particular, D is not polynomial-time computable ($D \notin \mathrm{P}$).*

命題 3 *Suppose that k is a positive integer, f is a function from $\{0,1\}^k$ onto $\{0,1\}$, and D is an oracle such that D is r-Dowd for all positive integer r. Then, there exists a natural number i such that*

$$f(D(z(i)), D(z(i+1)), \cdots, D(z(i+k-1))) = 0. \tag{6.1}$$

かくして，第 1 節の要請 2 はみたされる．

命題 4 *Suppose that r is a positive integer, A is an r-Dowd oracle, $f : \{0,1\}^* \to \{0,1\}$ is a polynomial-time computable function, and $T \subseteq \{0,1\}^*$ is a sparse set such that T is polynomial-time computable. Let B be an oracle defined as follows. For each string u, if $u \in T$ then $B(u)$ is defined as to be $f(u)$, and otherwise $B(u)$ is defined as to be $A(u)$.*

Then, B is r-Dowd.

よって，第 1 節の要請 3 はみたされる．また，性質 1 をもつオラクル D であって，なおかつ P-immune でないものが存在することがわかる．

定理 5 *Suppose that s and r are positive integers, A is an sr-Dowd oracle and that f is a function from $\{0,1\}^s$ onto $\{0,1\}$. Let B be an oracle defined as follows. For each natural number m, we define $B(z(m))$ as to be $f(A(z(sm)), A(z(sm+1)), \cdots, A(z(sm+s-1)))$.*

Then, B is an r-Dowd oracle.

系 6 *Suppose that m is a positive integer. Then, there exist oracles D_1 and D_2 such that each of them is an r-Dowd oracle for all positive integer r, and such that the following holds.*

$$\lim_{n\to\infty} \frac{|\{i \le n : D_1(z(i)) = 1\}|}{n} = \frac{1}{2^m}, \tag{6.2}$$

$$\lim_{n\to\infty} \frac{|\{i \le n : D_2(z(i)) = 1\}|}{n} = 1 - \frac{1}{2^m}. \tag{6.3}$$

よって，第1節の要請1はみたされる．

系 7 *Suppose that r is a positive integer, A is a $2r$-Dowd oracle, and B is an oracle defined as follows. For each natural number m, we define $B(z(m))$ as to be $A(z(2m))\,\mathrm{xor}\,A(z(2m+1))$.*

Then, B is an r-Dowd oracle.

アルゴリズム 3 の数学的モデル　オラクル・チューリングマシン M を次のように定める．

N, C_1, C_2 は正の整数とし，$C_3 = \mathrm{floor}(\log_2(C_1+C_2))$ とする．ただし，$C_2 < C_1 < N$ とする．また，N^2 を C_1 で割ったときの商を q_1 とする．オラクル X と入力 $z(k)$（ただし $k \in \mathbb{N}$）に対して，マシン M の出力 $M^X(z(k))$ を次のように定める．まず，k を q_1C_3 で割ったときの商を q_2，余りを k' とおく．

ビット列 $X(q_1C_1q_2)X(q_1C_1q_2+1)\cdots X(q_1C_1(q_2+1)-1)$ を a とおき，この a に対してアルゴリズム 3 の STEP 2 の操作を行う．次に，アルゴリズム 3 の STEP 3 の操作を行い，k' 番目に出力された値（0 あるいは 1 のいずれか）を $M^X(z(k))$ とする．

オラクルをオラクルに写す関数（汎関数）f を，上記のマシン M を用いて次のように定める．各々のオラクル X に対し，$f(X)$ はオラクル $\{u \in \{0,1\}^* : M^X(u) = 1\}$（の特性関数）であるとする．

このとき，オラクル X は（それが性質1を持つ限り）じゅうぶん多くのビットマップファイル（によってできる配列ないしはビット列）の数学的モデルであり，上記の関数 f はアルゴリズム 3 の数学的モデルである．このとき，以下が成り立つ．

定理 8 上記の関数 f は性質 *2* をもつ．

7 結び

本稿では，アルゴリズム 3 が乱数源の不規則性を保つことを定理 8 によって理論的に保証し，一方，アルゴリズム 3 の出力が統計的に見て「よい乱数」であることの確認は実験によって行った．

アルゴリズム 2 およびアルゴリズム 3 は，連に関するいくつかの検定において良好な結果を得ることができたが，それだけで精度の高い擬似乱数列を生成しているとはいいきれない．たとえば，乱数列のパターン性の検証については，本稿で行った検定だけでは不十分である．また，今回の実験に用いたサンプルの数は少なく，しかもすべて著者らが描いたものである．より多くのデータについて実験を行うべきである．

我々は，これらの課題に取り組むとともに，より精度の高い擬似乱数生成アルゴリズムの実装を行う．

また，性質2をもつ写像の具体例について，さらに理論を発展させたい．

謝辞 ビットマップファイルの扱い方について助言してくださった大阪府立大学 総合科学部 数理・情報科学科 の寶珍輝尚（Teruhisa Hochin）教授および，京都大学数理解析研究所での議論に参加してくださった各位をはじめとして，著者らに助言をくださった方々に，ここに謝意を表す．

参考文献

[AFH88] Ambos-Spies, K., Fleischhack, H. and Huwig, H.: Diagonalizations over deterministic polynomial time. In: *Proceedings of the first workshop on computer science logic, CSL '87*, Lecture Notes in Computer Science **329**, pp.1-16, Springer, 1988.

[Am96] Ambos-Spies, K.: Resource-bounded genericity. In: *Computability, enumerability, unsolvability*, London Math. Soc. Lect. Note Series **224** (S. B. Cooper, T. A. Slaman and S. S. Wainer, Eds.), pp.1-59, Cambridge University Press, Cambridge, 1996.

[AM97] Ambos-Spies, K., Mayordomo, E.: Resource-bounded measure and randomness. In: *Complexity, logic, and recursion theory*, Lecture Notes in Pure and Applied Mathematics **187** (A. Sorbi, Eds.), pp.1-47, Marcel Dekker, New York, 1997.

[ANT96] Ambos-Spies, K., Neis, H.-C. and Terwijn, S. A.: Genericity and measure for exponential time. *Theoret. Comput. Sci.*, **168** (1996), pp. 3-19. A preliminary version is appeared in: *Proceedings of the 19th international symposium on mathematical foundations of computer science, MFCS '94*, Lecture Notes in Computer Science **841**, pp.221-232, Springer, 1994.

[ATZ97] Ambos-Spies, K., Terwijn, S. A. and Zheng, X.: Resource bounded randomness and weakly complete problem. *Theoret. Comput. Sci.*, **172** (1997), pp. 195-207. A preliminary version is appeared in: *Proceedings of the 5th international symposium on algorithms and computation, ISAAC '94*, Lecture Notes in Computer Science **834**, pp.369-377, Springer, 1994.

[BDG88] Balcázar, J. L., J. Díaz, and J. Gabarró: *Structural complexity I.* Springer, Berlin, 1988.

[BDG90] Balcázar, J. L., J. Díaz, and J. Gabarró: *Structural complexity II.* Springer, Berlin, 1990.

[BG81] Bennett, C. H. and J. Gill: Relative to a random oracle A, $P^A \neq NP^A \neq co\text{-}NP^A$ with probability 1. *SIAM J. Comput.*, **10** (1981), pp. 96-113.

[Ca02] Calude, C.S.: *Information and Randomness: An Algorithmic Perspective, 2nd ed.*. Springer, Berlin / Tokyo, 2002.

[Do92] Dowd, M.: Generic oracles, uniform machines, and codes. *Information and Computation*, **96** (1992), pp. 65-76.

[Fe68] Feller, W.: *An introduction to probability theory and its applications I (third edition)*. John-Wiley, New York, 1968.

[Fe71] Feller, W.: *An introduction to probability theory and its applications II (second edition)*. John-Wiley, New York, 1971.

[Ge03] Gentle, J. E.: *Random number generation and Monte Carlo methods (second edition)*. Springer, New York, 2003.

[Go99] Goldreich, O.: *Modern cryptography, probabilistic proofs and pseudorandomness*. Springer, Berlin / New York, 1999.
邦訳 O. ゴールドライヒ著 ; 岡本龍明, 藤崎英一郎 訳『現代暗号・確率的証明・擬似乱数』シュプリンガー・フェアラーク東京, 東京, 2001.

[Kn98] Knuth, D.E.: *The art of computer programming vol.2, third edition.* Addison-Wesley, 1998.

[LV97] Li, M. and Vitányi, P., *An introduction to Kolmogorov complexity and its applications(second edition)*, Springer, New York, 1997.

[Lu92] Lutz, J. H.: Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, **44** (1992), pp. 220-258.

[Ml66] Martin-Löf, P.: The definition of random sequences. *Information and Control*, **9** (1966), pp. 602-619.

[NTs99] Nisan, N. and Ta-Shma, A.: Extracting randomness: a survey and new constructions. *Journal of Computer and System Sciences*, **58** (1999), pp. 148-173.

[NZ93] Nisan, N. and Zuckerman, D.: More deterministic simulation in logspace. In: *Proceedings, 25th annual ACM symposium on the theory of computing*, ACM, 1993, pp. 235-244.

[NZ96] Nisan, N. and Zuckerman, D.: Randomness is linear in space. *Journal of Computer and System Sciences*, **52** (1996), pp. 43-52.

[PM88] Park, S. K. and Miller, K. W.: Random Number Generators: Good Ones Are Hard to Find. *Communications of the ACM*, **31** (1988), pp. 1192-1201.

[Sch71a] Schnorr, C. P.: A unified approach to the definition of random sequences. *Mathematical Systems Theory*, **5** (1971), pp. 246-258.

[Sch71b] Schnorr, C. P.: Zufälligkeit und Warscheinlichkeit. In: *Lecture notes in mathematics 218*, Springer, New York, 1971.

[Sh02] Shaltiel, R.: Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, **77** (2002), pp. 67-95.

[Su01] Suzuki, T.: Forcing complexity: minimum sizes of forcing conditions. *Notre Dame J. Formal Logic*, **42** (2001), pp. 117-120.

[Su02] Suzuki, T.: Degrees of Dowd-type generic oracles. *Inform. and Comput.*, **176** (2002), pp. 66-87.

[Su05] Suzuki, T.: Bounded truth table does not reduce the one-query tautologies to a random oracle. *Archive for Mathematical Logic*, to appear.

[SY04] Suzuki, T. and Yamakami, T.: Resource bounded immunity and simplicity, extended abstract. In: *Exploring New Frontiers of Theoretical Informatics* (Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science, August 23-26, 2004, Toulouse, France; Levy et al. Eds.), pp.81-95, Kluwer Academic, 2004.

[vL90] Van Leeuwen, J. ed.: *Handbook of theoretical computer science, volume B: Formal Models and Semantics.* Elsevier, Amsterdam / Tokyo (the same book is published by MIT press, Cambridge / Massachusetts, too), 1990. 邦訳 Jan van Leeuwen 編 ; 廣瀬 健, 野崎 昭弘, 小林孝次郎 監訳『コンピュータ基礎理論ハンドブック 第2巻 形式的モデルと意味論』丸善, 東京, 1994.

[YDD04] Yu, L. and Ding, D. and Downey, R.: The Kolmogorov complexity of random reals. *Annals of Pure and Applied Logic*, **129** (2004), pp. 163-180.

[Zuc90] Zuckerman, D.: General weak random source. In: *31st Annual IEEE symposium on foundation of computer science, vol. II*, pp.534-543, IEEE, 1990.

[石井 95] 石井博昭, 塩出省吾, 新森修一『確率統計の数理』, 裳華房 (1995).

[鈴木 99] 鈴木登志雄『クエリー記号付きブール式のフォーシング計算量』, 日本数学会 1999 年度 秋季総合分科会 於 広島大学 ・数学基礎論分科会 特別講演 (1999). http://wwwmi.cias.osakafu-u.ac.jp/~suzuki/toshio_suzuki_msj99f.pdf

[伏見 89] 伏見正則『乱数』, 東京大学出版会 (1989).

[宮坂 04] 宮坂電人『プログラミング雑技談 第24回 BMP ファイル作成プログラム』, C MAGAZINE 2004 年 6 月号, pp113-117 (2004).

[脇本 70] 脇本和昌『乱数の知識』, 森北出版 (1970).

[和田 04] 和田維作『特集 1 乱数をきわめろ！』, C MAGAZINE 2004 年 6 月号, pp20-28 (2004).